## REMARKS

Applicant has studied the Office Action dated June 17, 2005 and has made amendments to the claims. It is submitted that the application, as amended, is in condition for allowance. By virtue of this amendment, claims 1-7, 9-14, 16, 17, 19-22, and 24-28 are pending. Claims 8, 15, 18, and 23 have been canceled without prejudice. Claims 1, 2, 13, 14, 16, 21, 22, and 24 have been amended, and new claims 25-28 have been added. Reconsideration and allowance of the pending claims in view of the above amendments and the following remarks are respectfully requested.

Claims 1, 2, 14, 15, 22, and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Pfab (U.S. Patent No. 6,195,752). Claims 3-13, 16-21, and 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Pfab in view of Menezes et al. ("Book of Applied Cryptography," pp. 10, 51, 64). Claims 8, 15, 18, and 23 have been canceled so, with respect to these claims, these rejections are moot. With respect to claims 1-7, 9-14, 16, 17, 19-22, and 24, these rejections are respectfully traversed.

The present invention is directed to methods and circuits for transferring data in a highly secure manner. One preferred embodiment provides a method for secured transfer through a data bus that is connected between a first memory and a second memory. According to the method, an N-byte data element is provided in the first memory, and the value of at least one parameter of a transfer rule is randomly chosen before a transfer of the N-byte data element. The transfer rule defines the order in which the bytes of the N-byte data element are successively transferred through the data bus. The N bytes of the data element are successively transferred byte-by-byte through the data bus to the second memory in the order specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus.

Thus, in this transfer method, the data element is stored in a first memory, the bytes of the data element are successively transferred byte-by-byte to the second memory, and the transfer rule with at least one randomly chosen parameter is used to select the byte to be transferred at

each successive transfer of a byte. The transfer rule, randomly chosen parameter(s), and successive byte-by-byte transfer of the data element operate together so that the N bytes of the data element are not successively transferred (through the data bus to the second memory) in the same order for each transfer of that data element. Accordingly, the simple power analysis method of snooping is not sufficient to obtain the value of the N-byte data element transiting through the data bus.

The Pfab reference is directed to a data processing circuit in which the data transiting on the data bus and stored in memory is protected by being encoded. However, Pfab does not disclose a method for secured transfer in which an N-byte data element is provided in a first memory, the value of at least one parameter of a transfer rule that defines the order in which the bytes of the data element are successively transferred is randomly chosen before a transfer, and the N bytes of the data element are successively transferred byte-by-byte through a data bus to a second memory in the order specified by the transfer rule, as is recited in amended claim 1. Amended claim 14 contains similar recitations.

Likewise, Pfab does not disclose a programmable circuit that includes a read-only memory containing an N-byte data element to be transferred, a random number generator that before a transfer supplies the value of at least one parameter of a data transfer rule that defines the order in which the bytes of the data element are successively transferred, and a control unit that controls a data bus such that the N bytes of the data element are successively transferred byte-by-byte through the data bus to a writeable memory in the order specified by the data transfer rule, as is recited in amended claim 22.

Pfab discloses a data processing circuit in which data is stored in memory and transferred through the data bus in an encoded format. In the first and second embodiments, the data processing circuit includes a microprocessor 101, a data bus 106, and memories 102-105, as shown in Figures 1 and 2. Each memory 102-105 stores encoded data, and this encoded data is transferred through the data bus 106. The microprocessor 101 includes an encoding module 107

that decodes the encoded data received from the data bus 106, and encodes data to be sent each memory 102-105 through the data bus 106.

In the third embodiment, the data processing circuit includes a microprocessor 1, data buses 6-15, and memories 2-5, as shown in Figure 3. Each memory 2-5 stores encoded data, and this data is transferred through the data buses 6-15 at least partially encoded. The microprocessor 1 includes one encoding module 35 and an additional encoding module 20-22 is provided on the data buses 6-15 between each memory 2-5 and the microprocessor 1. The encoded data stored in each memory 2-5 is partially decoded by the associated encoding module 20-22 and then completely decoded by the encoding module 35 of the microprocessor. Similarly, data sent to each memory 2-5 is partially encoded by the encoding module 35 of the microprocessor and then completely encoded by the associated encoding module 20-22.

Thus, Pfab teaches data processing circuits in which data is protected by modifying (i.e., encoding) each byte of data stored in memory and transferred through the data bus. While Pfab teaches modifying each data unit (e.g., byte) that is transferred, Pfab does not teach or suggest modifying the order in which these data units (e.g., bytes) are successively transferred through the data bus.

In contrast, in embodiments of the present invention, the N bytes of a data element are successively transferred byte-by-byte through a data bus using a transfer rule with one or more randomly chosen parameters so that the N bytes of the data element are not successively transferred in the same order for each transfer of that data element. More specifically, an N-byte data element is provided in a first memory. A transfer rule defines the order in which the N bytes of the data element are successively transferred through a data bus, and the value of one or more parameters of the transfer rule are randomly chosen before a transfer of the data element. The N bytes of the data element are successively transferred byte-by-byte through the data bus to a second memory in the order specified by the transfer rule, with each of the N bytes transiting once and only once through the data bus.

Thus, the data element is stored in a first memory, the bytes of the data element are successively transferred byte-by-byte to the second memory, and the transfer rule with at least

one randomly chosen parameter is used to select the byte to be transferred at each successive transfer of one of the N bytes of the data element. This causes the bytes of the data element to not always be successively transferred through the data bus in the same order.

Examples will now be given to highlight the differences between Pfab and embodiments of the present invention. In each example, we will successively transfer three data units B1, B2, and B3. On the bit level, these data units are represented as: B1 = "a1 b1 c1 d1 e1 f1 g1 h1", B2 = "a2 b2 c2 d2 e2 f2 g2 h2", and B3 = "a3 b3 c3 d3 e3 f3 g3 h3".

In Pfab, each data unit (e.g., byte) is transferred in a modified form, but these data units (e.g., bytes) are always successively transferred through the data bus in the same order. Pfab teaches interchanging bit lines of the data bus to modify the data units that are transferred. In this example, for this transfer of the three data units the bit lines of the data bus are interchanged so a data unit "abcdefgh" becomes encoded to "bdghceaf". Given the three data units B1, B2, and B3, in this example, these data units are successively transferred through the bus in the following order: first B1, then B2, then B3. However, the bits of each data unit are mixed up. Thus, these three data units would be transferred by first sending through the data bus "b1 d1 g1 h1 c1 e1 a1 f1", then sending through the data bus "b2 d2 g2 h2 c2 e2 a2 f2", and then sending through the data bus "b3 d3 g3 h3 c3 e3 a3 f3". Thus, each data unit is modified, but the order in which the data units are successively transferred is always the same.

In embodiments of the present invention, the data units are not always successively transferred through the data bus in the same order. In particular, the data units are successively transferred unit-by-unit through the data bus using a transfer rule (with one or more randomly chosen parameters) that defines the order in which the data units are successively transferred through the data bus. In this example, the transfer rule is: X = (X0 + DIRECTION * PITCH * j) modulo N, with the parameter PITCH chosen randomly. The other two values are constants: X0 = 2 and DIRECTION = 1. Before a first transfer, the value of PITCH is randomly chosen as 1, so the transfer rule is written as "X = (2+j) modulo 4". Therefore, on the first transfer the data units are successively transferred through the data bus in the following order: first B2, then B3, and then B1. Then, before a second transfer, the value of PITCH is randomly chosen as 2, so the

transfer rule is written as "X = (2+2*j) modulo 4". Therefore, on the second transfer the data units are successively transferred through the data bus in the following order: first B2, then B1, and then B3. Thus, the order in which the same data units are successively transferred is not the same at each transfer.[1]

Pfab only teaches modifying each data unit that is transferred, and does not teach or suggest modifying the order in which data units are transferred. Nowhere does Pfab teach or suggest a circuit or method for successively transferring an N-byte data element byte-by-byte through a data bus to a second memory while varying the order in which these same N bytes are transferred.

Embodiments of the present invention use a transfer rule, randomly chosen parameter(s), and successive byte-by-byte transfer of an N-byte data element such that the N bytes of the data element are not successively transferred in the same order for each transfer of that data element. Accordingly, the simple power analysis method of snooping is not sufficient to obtain the value of the N-byte data element transiting through the data bus. Thus, in embodiments of the present invention, data is protected by modifying the order in which the bytes of a data element are successively transferred byte-by-byte, not by modifying the bytes themselves. In other words, regardless of whether the bytes of the data element are stored in memory and transferred in clear or encoded format, the order of successive transfer of the bytes varies.

---

[1] These two examples hold true regardless of the size of each data unit or the number of data units to be transferred. In embodiments of the present invention, the size of each data unit is a byte, so "byte" can be substituted for "data unit" in the above example for the present invention. In the above example for Pfab, "byte" can also be substituted for"data unit" for ease in identifying the differences. However, any size "data unit" can be used in the above example for Pfab and the same difference is still present, because embodiments of the present invention require the unit-by-unit successive transfer of the data units through the data bus. In other words, regardless of data unit size in either example, Pfab modifies each data unit but not the order in which the data units are successively transferred, while embodiments of the present invention vary the order in which the data units are successively transferred.

Applicant believes that the differences between Pfab and the present invention are clear in amended claims 1, 14, and 22, which set forth various embodiments of the present invention. Therefore, claims 1, 14, and 22 distinguish over the Pfab reference, and the rejection of these claims under 35 U.S.C. § 103(a) should be withdrawn.

As discussed above, amended claims 1, 14, and 22 distinguish over the Pfab reference. Furthermore, the claimed features of the present invention are not realized even if the teachings of Menezes are incorporated into Pfab. Menezes does not teach or suggest the claimed features of the present invention that are absent from Pfab. Thus, claims 1, 14, and 22 distinguish over the Pfab and Menezes references, and thus, claims 2-7 and 9-13, claims 16, 17, and 19-21, and claim 24 (which depend from claims 1, 14, and 22, respectively) also distinguish over the Pfab reference. Therefore, it is respectfully submitted that the rejections of claims 1-7, 9-14, 16, 17, 19-22, and 24 under 35 U.S.C. § 103(a) should be withdrawn.

Claims 25-28 have been added by this amendment, and are provided to further define the invention disclosed in the specification. Claims 25-28 are allowable for at least the reasons set forth above with respect to claims 1-7, 9-14, 16, 17, 19-22, and 24.

In view of the foregoing, it is respectfully submitted that the application and the claims are in condition for allowance. Reexamination and reconsideration of the application, as amended, are requested.

If for any reason the Examiner finds the application other than in condition for allowance, the Examiner is invited to call the undersigned attorney at (561) 989-9811 should the Examiner believe a telephone interview would advance the prosecution of the application.

Respectfully submitted,

Date: December 19, 2005

By: _____
Stephen Bongini
Registration No. 40,917
Attorney for Applicant

FLEIT, KAIN, GIBBONS,
GUTMAN, BONGINI & BIANCO P.L.
One Boca Commerce Center
551 Northwest 77th Street, Suite 111
Boca Raton, Florida 33487
Telephone: (561) 989-9811
Facsimile: (561) 989-9812